Social-Aware Video Pre-fetching for Multimedia Streaming

Nikhil N. Katore
PG Student: Department of I.T.
Sinhgad College of Engineering,
Pune, India
E-mail: nikhil.katorenk@gmail.com

Dr. Nilesh J. Uke
Professor & Head: Department of I.T.
Sinhgad College of Engineering,
Pune, India
E-mail: nilesh.uke@gmail.com

Abstract— **As the demand of multimedia streaming services over the mobile networks have been soaring over these years. But, the capacity of wireless links cannot keep up with the increasing traffic load. The increasing gap between traffic demand and the link capacity with time-varying link conditions arises to worthless service quality of multimedia streaming, such as long buffering delays and intermittent disruptions. In this paper, by leveraging the present cloud computing technology, we proposed AMES-cloud architecture. Cloud-assisted adaptive video streaming and social-aware video prefetching are the two basic parts of the proposed system. The separate private agent is constructed in the cloud center for each active mobile user to adaptively rectify video bit-rate (quality) by using scalable video coding technique. Private agent also monitors the social interaction among mobile users; so that each mobile user can share their videos through the private agents which will be effectively prefetched (based on the social network analysis) to other mobile users in advance. The proposed approach will enhance the adaptability of the multimedia streaming and the effectiveness of social-aware prefetching supported by cloud computing environment. Meanwhile, this also reduces the access delay of each video which is shared among mobile users.**

Keywords- Adaptive Video Streaming; Scalable Video Coding; Mobile Cloud Computing; Social Video sharing; Pre-fetching.

I. INTRODUCTION

In recent years, many video service applications over the Internet have gained popularity, in which the video streaming is one of the most successful video services. As a result, the traffic estimated from video, excluding games and shared files, will reach 55% of the total Internet traffic in 2016. The video streaming technology is now mature enough to offer such services to the end-users over the Internet. The era of cloud computing reigns with advancements in technology that can offer various services to the users and also it urges the more necessity for the emerging technology. Cloud computing provides a platform for other advanced technologies like mobile computing and big data to instil its services and provide the QoS to the users [1].

Video streaming is getting a frequent part of user's daily life on the Web, with the essential development of internet and communication technology. Many people are attracted to enjoy video streaming services through different networks and different terminals, which directly resulted in the heterogeneity of the multimedia communication network environment. The terminals of the users of multimedia communication are different, such as various types of mobile phones, PDA, tablet PCs, etc. These terminals are different in processing capability, display resolution and power consumption, which directly represents the diversity of multimedia communication terminals. Video streaming services should support a large range of mobile devices because they have different video resolutions, different computing powers, and different wireless links (3G, Wi-Fi and 4G). In spite of the great efforts of network operators to enhance the wireless link bandwidth, the rising video traffic demands of mobile users are rapidly overwhelming the wireless link capacity. The main issues faced during the study of video streaming and sharing achieved in mobile users under cloud environment are higher traffic rate, greater buffering delay, and disruption due to the bandwidth variation and link fluctuation. Many issues had occurred and resolved through various techniques during the traditional change happened between emerging technologies.

The recent research shows that various services are provided by the service provider to the users, which must offer a good quality of services (QoS) for mobile video streaming while utilizing the networking and computing resources efficiently. Therefore service providers re-thinking what the core QoS factors for mobile video streaming and how to design new techniques for better performance. So that mobile users can enjoy stable and continuous video streaming without disruptions. Also enjoy click-to-play video streaming with less buffering delays for satisfying the user's expectation so as to offer good quality of experience (QoE).

Multimedia streaming services over mobile networks have been increasing over these years. Despite the network (like 3G/4G & Wi-Fi) enhances the wireless link capacity, but still it cannot practically keep up with the increasing traffic load. Due to this gap between link capacity and traffic demands along with the time varying link condition results to poor quality of multimedia streaming services over the mobile networks. The variability in the bandwidth and link fluctuation arises various problems such as slow startup time, long buffering delays and intermittent disruptions. The goal is to offer a good quality of services (QoS) for multimedia streaming while utilizing the networking and computing resources efficiently. So that mobile users can enjoy stable and continuous video streaming without disruptions. Also enjoy click-to-play multimedia streaming with less buffering delays for satisfying the user's expectation so as to offer good quality of experience (QoE) and reliable services to users. Hence we are motivated to explore the relationship between different mobile users from their social networking sites (SNSs) activities in order to pre-fetch the multimedia contents. Prefetching will fetch the multimedia contents in advance to the user device before user access it. It provides the efficient multimedia prefetching and sharing among the various users and also provides adaptive multimedia streaming services.

Over the past two decades, there have been many related research studies on how to improve the QoS and QoE of mobile video streaming which focusing on two aspects:

A. Adaptability:

Mostly designed traditional video streaming techniques uses the stable Internet links between servers and users and hence they may perform poorly in mobile environments. Those works with a particular or fixed bit rate, the video streaming can be frequently disrupted due to packet losses if the link bandwidth varies much. Hence, the fluctuating link condition should be properly handled to provide more stable video streaming service. So as to offer a better QoS experience by which the video quality can adapt to the environment. To notice this issue, the time-varying available link capacity of each mobile user is tuned with the video bit rate adaption is used, and it based on user feedback of the link quality [2].

B. Scalability:

Mobile multimedia streaming services should also support a wide spectrum of mobile devices, with different screen resolutions, different power supplies, and different wireless accesses (e.g., Wi-Fi, 3G and 4G). However, the traditional method to store multiple versions of the same video content may create a large amount of storage overhead while the volume of video content is skyrocketing globally. The Scalable Video Coding (SVC) technique of the H.264/AVC video compression standard, which describes a base layer (BL) with multiple enhancement layers (ELs). By utilizing the SVC, a video can be decoded and displayed at the lower quality if only the BL is delivered. While the more ELs are delivered to achieve better quality of the video stream. Therefore a high diversity of mobile devices and link conditions can be covered scalably [3].

Most of current research proposals that seek to combine the video scalability and adaptability highly rely on the active control from the server side. That is, all mobile users individually report the transmission status periodically to the server, and the server predicts the available bandwidth and allocates proper video streams for each user.

In this paper, we focus on improving the response time and access delay while watching the videos. The proposed cloud architecture will enhance the adaptability of multimedia streaming and effectiveness of social aware video prefetching. This reduces the effect of low bandwidth, intermittent disruption and low wireless link capacity. Meanwhile, user can share the videos more efficiently over the SNSs.

The remainder of this paper is organized as follows: Section 2 discusses some related work. Section 3 presents AMES-Cloud architecture. Section 4 discusses about streaming flow and video storage by AMoV and EMoS. Section 5 discusses about the implementation and evaluation. Section 6 concludes the paper and discusses future work.

II. RELATED WORK

A. Adaptive Video Streaming Techniques:

In adaptive video streaming the video traffic rate is well-balanced so that a user can experience the maximum achievable video quality [2]. Adaptive streaming techniques are mainly classified into two types depending on adaptability is triggered by client or server. The work in [7], proposes Microsoft's Smooth Streaming can switch between different bitrate segments encoded with configurable bitrates as well as video resolutions at servers. In addition, clients can dynamically request videos based on local monitoring of link quality.

In [3], MPEG-21 Digital Item Adaptation (DIA) is integrated with SVC for an efficient adaptation framework and it is shown that SVC can seamlessly be adapted using DIA. DIA enables interoperable access to distributed (advanced) multimedia content by shielding users from the network and terminals. However, most of these solutions preserve large number of copies of the multimedia content with various bit rates, which leads to a huge burden of storage on the server.

B. TCP-friendly rate control (TFRC):

Concerning the rate adaptation controlling methods TCP-friendly rate control methods for multimedia streaming services over mobile networks are proposed in [8], [9]. Where the bit rate of the stream can be dynamically adapted to the changing channel conditions which greatly improve the performance indicators such as packet loss rate, interruption time, round trip time, packet size, delay and buffer requirements through which TCP throughput of a flow is predicted. This also signifies that more users could be admitted to the cell and still it would be able to guarantee certain service qualities. This is especially true in a loaded situation where there are not sufficient radio resources to resist bad reception quality to some users. The bit rate of the streaming traffic can be adjusted by considering the estimated throughput. In [10], for conversational 3G video streaming simplified rate adaptation algorithm is introduced. The work in [11], [12], make the rate adaptation more accurately by few cross-layer adaptation methods, which can acquire more precise information regarding link quality. Though the servers have to always control and thus suffer from large workload. However, since the TFRC was not designed for a mobile environment so it is expected that it can be further optimized.

Recently, the H.264 SVC technique has gained a popularity [4], since the primary objectives of ongoing research on SVC are to significantly achieve high compression efficiency, high flexibility (i.e., Bandwidth scalability), and low complexity. But the minor losses in coding efficiency are possible when the application requires low delay. An adaptive video streaming system based on SVC is proposed in [3], which studies the real-time SVC implementation and includes provisions for protecting SVC transmission in an error prone RTP environment by decoding and encoding at PC servers. This provides an unequal erasure protection scheme for SVC to protect the packet losses in an error prone environment. In [13], a quality-oriented and scalable video delivery is offered by using H.264 SVC to assess the video quality of scalable video streamed over an LTE network through the use of multiple objective quality metrics such as SSIM, PSNR, Blocking and Blurring. Concerning with the encoding performance of SVC, CloudStream mainly offers the delivery of high-quality streaming videos by transcoding the original video in real time to a scalable codec through a cloud-based SVC proxy [5]. This allows streaming adaptation to network dynamics and discovered that the cloud computing can expressively enhance the performance of SVC coding. The use of SVC for video streaming on top of cloud computing is promoted by above studies.

C. *Mobile Cloud Computing Techniques:*

The cloud computing techniques is used to flexibly provide scalable resources to service providers, content and process offloading to mobile users [14]. So, the cloud data centers can easily provision of large scale real-time multimedia services as introduced in [3], [5]. Cloud computing has been well positioned to offer video streaming services, especially in the wired Internet on the basis of its scalability and capability [15]. As an example, the work in [16] proposes the quality-assured bandwidth auto-scaling system for video on demand (VoD) streaming based on cloud computing. That allows a predictive resource auto-scaling, which dynamically books the minimum bandwidth resources from multiple data centers for the VoD provider to match its short-term demand projections. As well as the CALMS framework [17] provides a cloud-assisted live media streaming services with time or region diversities for globally distributed users. There are many factors that need to be considered for extending the cloud computing based services to mobile environments such as user mobility, wireless link dynamics and the limited capability of mobile devices [18], [19].

More recently, several studies have provided novel plans for users on the basis of mobile cloud computing environments have proposed to generate personalized, intelligent private agents that are in charge of satisfying the demands of each user such as Cloudlets [20] and Stratus [21]. This is because of the cloud, in which multiple agent instances (i.e., Threads) can be maintained dynamically depending on the time-varying user requirements.

As the popularity of social network services (SNSs) have been increasing over these years. There have been proposals to enhance the quality of media delivery by using SNSs. In which users may comment, share or re-post videos among friends as well as members of the group, which means that a user may watch a video that his/her friends have recommended. Users in SNSs can also follow popular and famous users based on their interests, which is probably to be watched by its followers [22], [23].

Unlikely the traditional Really Simple Syndications (RSS) PC software, the appropriate user interface and fast response speed on mobile devices become very important. In addition, helping the user for finding interesting news and filter out the unnecessary ones for mobile users become more critical. The work in [24] studies the recommendation features for an RSS reader, as well as the [25] is also about an intelligent RSS reader for mobile devices by which the users can access to interested or popular content more easily. To find good news for users based on the rating of other users, the Google News application also uses online collaborative filtering [26]. Furthermore, many researchers design to push the media content to the mobile users based on their RSS subscriptions such as publish-subscribe middleware in [27] and the user preference-based pushing in [28]. Due to the development of cloud computing technology, pushing by the cloud is gaining more attention as a result the cloud-based push-styled mobile content dissemination system is shown in [29].

III. SYSTEM ARCHITECTURE

This section gives detailed view about the proposed AMES-cloud architecture, which offers the adaptive and efficient way of enhancing the multimedia streaming and sharing of videos for the mobile users. This architecture includes and provides services under two main methodologies adaptive mobile video streaming (AMoV) and efficient social video sharing (ESoV).

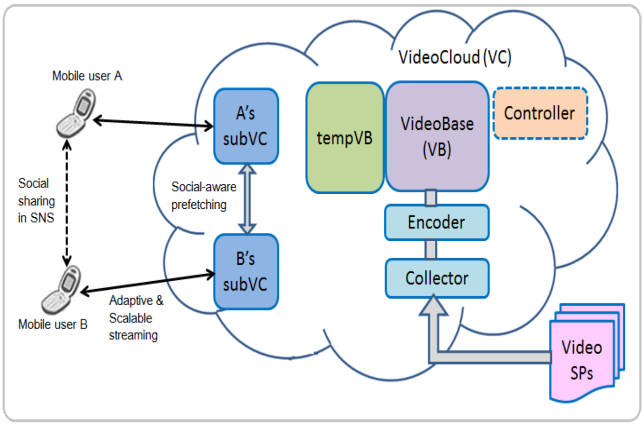

Figure 1. AMES-Cloud Architecture using Video Cloud (VC)

As shown in Fig. 1, the Video Cloud (VC) is the whole multimedia storing and streaming system in the cloud. There is a large amount of video base (VB) in the VC, which collects and stores the most popular multimedia data from the video service providers (VSPs). VC also has a temporal video base (tempVB) which is used to cache new users for the popular videos, while the tempVB mainly works to count the access frequency of each video. Most popular videos are stored in VB. A collector is used to seek the videos which are already popular in VSPs; hence VC keeps running a collector. The collected videos are re-encoded into SVC format and then stored into tempVB. The AMES-Cloud can keep offering most popular videos continually by using this 2-tier storage. The whole management work will be handled by the collector in the VC.

Especially for each mobile user, a private agent known as a sub-video cloud (subVC) is dynamically created if there are any multimedia streaming demands of the user. For storing recently fetched video segments the subVC has a sub video base (subVB). Note that mostly the deliveries among the subVC's and the VC are actually not copy, but just link operations on the same file continually within the cloud data center. The subVC also has an encoding function with an encoder instance of the encoder in VC, when the user requests for a new video, which is not stored in subVB or the VB in VC, then subVC will bring, encode and transfer the requested video form the outside video service providers. During multimedia streaming mobile users will frequently report link conditions to their respective subVC's, and the subVC's offers adaptive multimedia streams. The important part which needs to be focused is the catching storage of the each mobile device; this temporary storage is called as local video base (localVB), which is used for prefetching and buffering.

As the cloud may provide services across different places or even continents, hence in that case the video delivery and prefetching between different data centers will be carried out by transmitting "copy" of content not just the "link". As the cloud data centers have the capable links among them, as well as the optimal deployment of data centers, the copy of a large video file takes small delay [30].

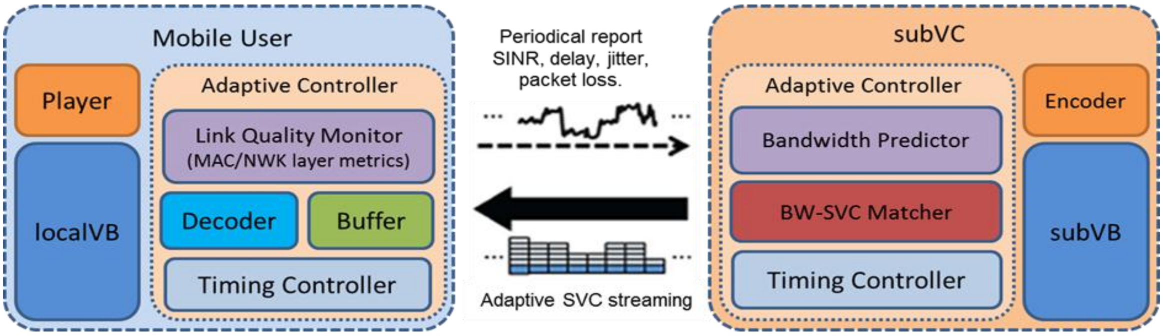

Figure 2. Functional structure of the mobile user and private agent (subVC)

A. Adaptive Mobile Video Streaming (AMoV):

There are several methods through which adaptive streaming can be achieved:

1) Scalable Video Coding (SVC):

Traditional video streams with fixed bit rates, which cannot adjust to the variations in the link quality. If the sustainable link bandwidth varies much for a particular bit rate, then the video streaming can be terminated many times because of the packet loss. While in the scalable video coding (SVC) there is a combination of the lowest scalability is called Base Layer (BL) and the improved associations are called Enhancement Layers (ELs). By using the SVC, if only the BL is delivered, it guarantees that video can be decoded and displayed at the lowest quality, while to achieve the best quality of the video streams the more ELs are delivered.

The server doesn't require concerning the client side or the link quality while using the SVC encoding method. Even in poor link quality and some packet loss still the client can decode and display the videos. Despite this capability, because of the unnecessary packet loss it is not bandwidth-efficient. So it is essential to control the SVC based multimedia streaming at the server side with the use of rate adaptation techniques for utilizing the bandwidth efficiently.

2) Adaptability with Monitoring on Link Quality:

As shown in Fig. 2, the design of the mobile client and their private agents (i.e. subVC) with the functional structure. The mobile clients keep track of the metrics such as packet round trip time (RTT), signal strength, jitter and packet loss with the certain duty cycle for monitoring the link quality. And the mobile client will periodically report to the subVC. Here the cycle period for reporting is known as "time window". Note that the video can split into segments on the basis of time window which is denoted by "T_{win}".

When the subVC gets the information about the link quality it will predict the potential bandwidth in the next T_{win} by performing some calculation. Remark that we will use the "predicted bandwidth" and "predicted good-put" alternately in further part.

Suppose 'i' is the sequence number of the current T_{win} then the predicted bandwidth can be calculated by:

$$BW_{i+1}^{estimate} = BW_i^{practical}.[\alpha.f(p_i,p_{i-1}) + \beta.g(RTT_i,RTT_{i-1}) + \gamma.h(SINR_i,SINR_{i-1})]$$

Where, $\alpha+\beta+\gamma=1$ showing the importance of the each factor, p indicate packet loss rate, RTT is for round trip time, for indicating the signal to interference and noise ratio SINR is used. There are three functions f(), g(), h() that reflects the value change of each factor as compared to the last T_{win}.

As a matter of fact, this paper displays a measurement-based prediction, since we directly use $BW_{Practical}$ of last T_{win} as $BW_{estimate}$ of next T_{win}, which is already proven with highest accuracy [31].

3) Matching Between Bandwidth Prediction and SVC Segments:

Once the predicted bandwidth or good-put of the next time window is obtained, subVC will match and determine that nearly how many numbers of video segments of Base Layer (BL) and ELs can be transmitted.

As SVC define different profiles of video streams with one BL and multiple ELs. These layers or say sub streams can be encoded by utilizing following three scalability features:

a) *Spatial scalability:* This can be achieved by layering image resolution (i.e. Screen pixels)
b) *Temporal scalability:* This can be achieved by layering the frame rate
c) *Quality scalability:* This can be achieved by layering the image compression and thus can offer videos for a high variety of quality with relatively less storage overhead.

The term “resolution” is used to designate the level of temporal segmentation as well as the number of ELs offered to the user. As video is segmented by the T_{win}, if it is small and there are more ELs offered, then consider that SVC based video source has a higher resolution. To matching among the SVC segments and the predicted good-put there are two cases of coarse-grained (low resolution) and a fine-grained (high resolution). The resolution with two ELs and a larger T_{win} can hardly fit to the signal variation and thus there are packets lost or some bandwidth wasted. Hence, more ELs are needed for the higher resolution and a smaller can always fit the fluctuation of the bandwidth. But a higher resolution also influences the more workload to servers for encoding.

Consider there is total *j* ELs, while the bit rate of the jth EL is indicated as REL_j. That of the bit rate of the BL is RBL. We Let BL_i show the SVC segment of Base Layer with the temporal sequence *i*. Therefore, the algorithm of matching between predicted bandwidth and SVC segments is shown in Algorithm 1 as follows:

Algorithm 1. Bandwidth Prediction Algorithm

$i = 0$

$BW_0 = R_{BL}$

$Transmit\ BL_0$

Monitor $BW_{i+1}^{Practical}$

repeat

Sleep for T_{win}

Obtained p_i , RTT_i , $SINR_i$ etc., from client’s report

Predict

$BW_{i+1}^{estimate}\ (or\ BW_{i+1}^{estimate} = BW_i^{Practical})$

k = 0

$BW_{EL} = 0$

repeat

k ++

if k >= j break

$BW_{EL} = BW_{EL} + R_{EL^k}$

until $BW_{EL} \geq BW_{i+1}^{estimate} - R_{BL}$

Transmit $BL_{i+1} and\ EL_{i+1}^1, EL_{i+1}^2, \dots\dots, EL_{i+1}^{k-1}$

Monitor $BW_{i+1}^{Practical}$

i++

until All video segments are transmitted

B. Efficient Social Video Sharing (ESoV):

1) Social Content Sharing:

In social networking sites (SNSs), users can subscribe to known friends, famous people as well as to the particular interested content publishers. As shown in Fig. 3, there are ample types of social activities between the different users in SNSs, such as direct recommendation massage and public sharing.

To spread videos over the SNSs, user can post a video in the public, so that his/her subscribers can immediately see it. Also user can send a video to specified friends by direct recommendation. Furthermore user can periodically get notifications from subscribed content publisher for popular or new videos.

Similar to different investigation in [22], [23], there are different strength levels for those various social activities to show the possibility that the video shared by any user that may be watched by the recipients of one’s sharing activities, which is known as the “hitting probability”. Thus the subVC’s can perform the effective background prefetching at subVB as well as at the localVB. Since after sharing a video there may be a certain delay for the recipient that he could know the sharing and starts to watch the video [32]. Therefore, in most of the cases prefetching the contents before will not impact the users. Rather as the starting part of the video or even the complete video is already pre-fetched at the localVB of the user’s device still user can watch the video without any buffering delay. How much amount of video is pre-fetched is mainly determined by the type of the user’s social activities. While the prefetching from Video Cloud to subVC only refers to the “linking” actions, as there is only file locating and linking operations with small delays. The prefetching from the subVC to user’s localVB also depends on the strength of the social connectivity’s or activities as well as considers the wireless link status.

There are different social activities in current popular SNSs, which are classified into three kinds, considering possible reacting priority from the recipients’ point of view:

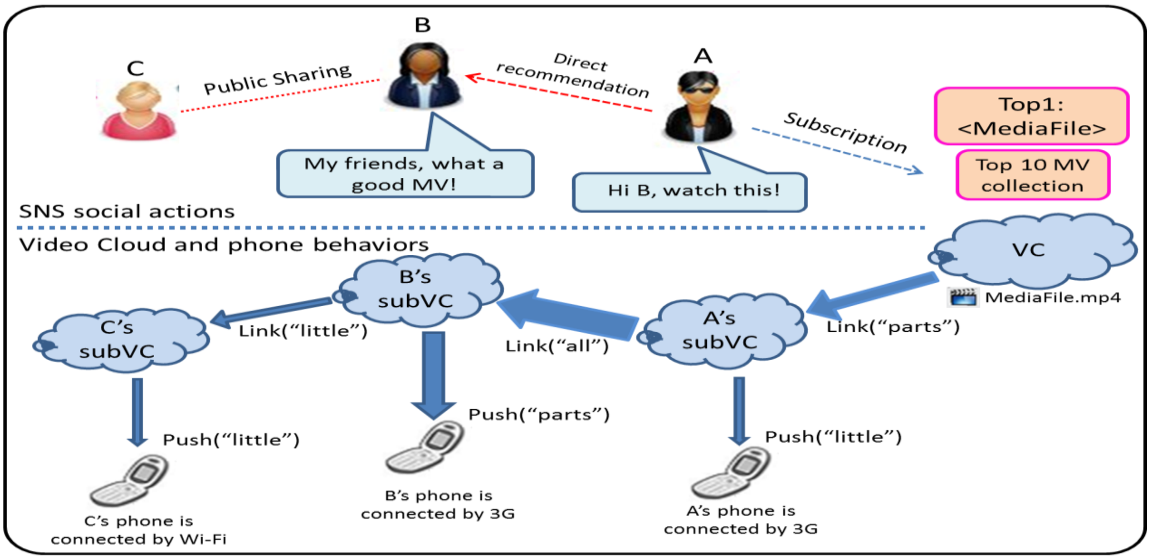

Figure 3. Social-aware multimedia prefetching

a) *Subscription:* Any user can subscribe to a special video collection service or a particular video publisher based on their interests. This interest based connectivity among the subscriber and the video publisher is considered as "median", since the subscriber may not always watch all subscribed videos.

b) *Direct recommendation:* In SNSs, any user can directly recommend or share a video to his/her friends with a short message. This is considered as "strong", because the probability is very high that the recipients of the message may watch a video.

c) *Public sharing:* In SNSs, every user has their timeline based on activity stream which indicates their recent activities. The act of watching or sharing a video by any user and that can be seen by their friends or followers. This public sharing considered as the "weak" connectivity among users, since not many people may watch the video that one has seen without the direct recommendation [6].

2) *Prefetching Levels:*

There are distinct levels of possibilities that a video will be soon watched by the recipient indicated by different strengths of the social activities. However, we define three prefetching levels concerning the social activities of mobile users:

a) *Parts:* In this to push only a part of BL and ELs segments are essential, because the probability is very less that the subscribers may watch the videos that are published by subscriptions.

b) *All:* The probability is very high that the recipient will watch the videos which are shared by the direct recommendations. Hence, it is essential to pre-fetch the BL as well as all ELs in order to let the recipients directly watch the video without any buffering delay and with a good quality.

c) *Little:* As the public sharing is considered to be a weak connectivity between the users, so the possibilities that user's friends or followers may watch the same video that has watched or shared by user is very low. Hence, it is essential to only pre-fetch the BL segment of the first T_{win} in the starting for those who have seen their activity in the stream.

TABLE I. SOCIAL ACTIVITIES AND BACKGROUND PREFETCHING

	Type of sharing		
	Direct recommendation	***Subscription***	***Public Sharing***
VB→subVB	All	Parts	Little
subVB→localVB (via Wi-Fi)	All	Parts	Little
subVB→localVB (via 3G/4G)	Parts	Little	None

All the prefetching which happens among subVB and the VB, and also more importantly, that will be performed from the subVB to localVB of the mobile device based on the wireless link quality. Mostly, if a mobile user has a Wi-Fi access then subVC can push as much as possible, due to Wi-Fi's link ability and low price cost. Though if the user has a 3G/4G connection, it's supposed to downgrade the prefetching level to preserve energy as well as the cost as listed in Table I, as it charges a lot and suffers limited bandwidth. But users can still advantages from the prefetching effectively.

Note that in order to actively decide whether the current battery status is suitable for "parts" or "little" there are some energy prediction methods can be deployed [35]. If a user known as A, gets the direct recommendation of a multimedia from another user, such as B then A's subVC will immediately pre-fetch the video either from B's subVB or from the VB (or tempVB) at the level of "all", if user A is with Wi-Fi access. But, if user A is associated to 3G or 4G link, then it will selectively pre-fetch a part of the video segment to A's local storage at the level of "parts". Remark that the subscribed videos will be not pre-fetched when user A is using 3G/4G connection as it is decreased from "little" to none.

A self-updating mechanism from the user's hitting history in an evolutionary manner, that can be designed a better way of the prefetching strategy is by social activities. However, this learning based multimedia prefetching is out of the scope of this paper which will be examined as our future work.

IV. Streaming Flow And Video Storage By AMoV AND EMoS

AMES-Cloud architecture has two parts, AMoV and EMoS these have tight connections and will together service the video streaming and sharing. Both of them is rely on the cloud computing platform and carried out by the user's private agents. The AMoV will still monitor and enhance the transmission considering the link status, while prefetching in EMoS, which prefetch certain amount segments and AMoV can exhibit better video quality.

The video will be streamed and accessed with the help of AMoV and EMoS with the particular flow. Note that a VMap (video map) is used to show the required segments, in order to interchange the videos between the subVBs, localVBs, VB and the tempVB.

Firstly, the localVB will checked whether there is any prefetched part of video data so that it can start directly, when a mobile user clicks to watch a video with a link. If there is just some part or none, then the client will report, a respective VMap to its subVC. The subVC will initiate the segment transmission, if the subVC has prefetched parts in its subVB. But if there is also no video is present in subVB then the video availability in tempVB as well as in VB of the VC will be checked. If the video does not exist in AMES-Cloud then the collector in the VC will immediately bring it from external video service providers via the link. The subVC will transfer the video to the mobile user after re-encoding it into the SVC format, by taking a bit longer delay.

Also in AMES-Cloud, when a video is shared between the subVC's at a certain level of frequency threshold (i.e., near about 10 times per day), then it will be moved and stored into the tempVB of VC. If the same video is furthermore shared by many users at a much higher frequency (i.e., more than 100 times per day), then it will be moved and uploaded permanently in the VB. In this manner the subVB and VB can always store popular and fresh videos in order to enhance the probability of re-usage; this is quite similar to the leveled CPU cache.

V. Implementation and Evaluation

A. *Experimental Setup:*

To evaluate the performance of the AMES-Cloud architecture using a prototype implementation by using the U-cloud server in the cloud computing service provided by Korean Telecom, and make utilize virtual server with 32 GB memory and 6 virtual CPU cores (2.66 GHz). This is fast enough for encoding 480 by 720 (i.e., 480P) video with H.264 SVC format in real time at 30 fps [3]. The server application based on Java has deployed in the cloud. This includes one main program handling all the tasks of the whole VC, during the program is dynamically initializes, maintains as well as terminates instances of any other Java application as private agents (subVC) for all active mobile users. Furthermore, we implement the mobile client with a mobile phone having the android system with version 4.0. In this the mobile data service is offered by LTE network or 3G network is used in some uncovered area. Remark that we still use 3G to show the general cellular network. The practical bandwidth of the wireless link is not as high enough as we expected, while testing in the downtown area. However, this will not impact our experimental results.

The test video is the Lagan Trailer in H.264 format with resolution of 480P downloaded from YouTube. Its duration is 180 seconds and with a size of 13.849 Mbytes. We first decode the video by the x264 decoder into the YUV format and then again encode it by using the H.264 SVC encoder with the software called Joint Scalable Video Model (JSVM) of version 9.1. For the decoding and encoding default settings are used, and perform the H.264 SVC encoding at the virtual server in the cloud.

B. *Adaptive Video Streaming Based on SVC:*

First of all we examine the relationship among the measured bandwidth (BW) of last T_{win} and practical bandwidth of the next T_{win} (good-put by Kbps), whether it is deep or not. After this the next test is the video streaming service through cellular link, and try to change the signal quality by moving the device around the building. This all tests are run five times. Then the relative error for the predicted BW to the practical BW for each T_{win} is collected. This is calculated by:

$$\left|BW^{estimate} - BW^{Practical}\right| / \{BW_{Practical}\}$$

Here, the bar denotes the 25% & 75% quartiles, and the whiskers denote 5% and 95% percentiles. The predicted bandwidth is very close to the practical bandwidth with around 10% relative error, when T_{win} is 1 second or 2 seconds. But the greater values of T_{win} have a relatively worthless prediction accuracy that reflects the similar results [31]. So for the accurate prediction in practical designs we indicate that a short T_{win} of 2or 3 seconds is required.

C. *Prefetching Delays:*

In ESoV the video segments (chunks) can be prefetched among tempVBs, VB and localVBs of the mobile users, depending on their activities in SNSs. The required delay for different levels of prefetching is evaluated. Here, the regular resolution configuration with 2 seconds temporal segmentation by default is used. And also set the sharing length of "little" as singly the starting's first 5 seconds of the BL and ELs, that of the first 15sec. of the BL and ELs as the "parts", and that of "all" as all BL and ELs segments.

The prefetching is significantly fast as it is supported by the cloud computing. While prefetching via wireless links takes several seconds. It is obvious that in most of the cases [32], [34] a recipient may not watch the shared video immediately after the original sharing behavior. As the normal users have expressive access delay gaps, hence this prefetching transmission delay will not affect the user's experience at all. At a later time when the user clicks and starts to watch the video, then it will bring "non-buffering" experience to the user.

D. *Watching Delay:*

By testing the normal time for which period user has to wait starting from the instant when he/she clicks the video in mobile device to the instant that the first streaming segment arrives to the user's device this is called as "click-to-play" delay. As shown in the Fig 4. The proposed system will helps to reduce this delay by prefetching the shared video to client's localVB, so that the video can be displayed immediately with an ignorable delay. However, when the user watches video which is accessed from the subVC or the VC, generally it takes less than 1 second to start. Instead, if the user accesses the AMES-Cloud services through the cellular link, then the user will still suffer a big longer delay (nearly 1s) due to the larger RTT of the transmission via the cellular link.

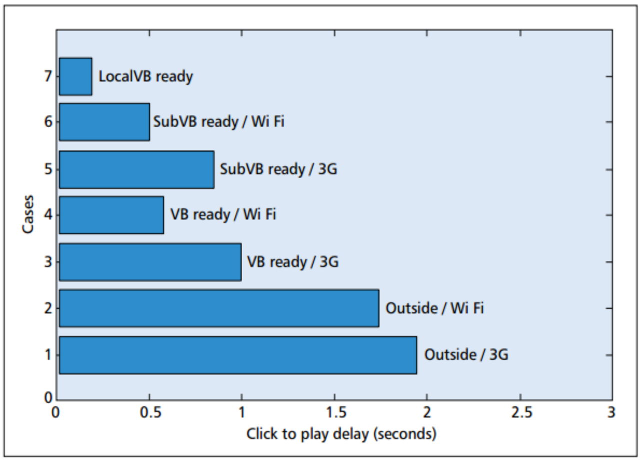

Figure 4. Click-to-play delays for various cases

If the fetch videos which is not stored in the cloud (but present at the video service provider's server), then the delay is a bit higher. This fetching delay is mainly due to accessing the video via the link from the VSPs to the cloud data center, and as well as the encoding delays in the cloud. But the access delays that analyzed recently are much higher than that click-to-play delay which means social-aware prefetching can perform perfectly to match the user demands. Recently this won't happen, since most of the popular videos will be already prepared in the cloud data center. Hence the accessing the videos via the proposed cloud system will give better user experience with reduced access delay compared to other VSPs. Further measurement and more tests will be carried out as future work.

VI. CONCLUSION AND FUTURE SCOPE

In this paper, in order to provide adaptive multimedia streaming and effective social-aware prefetching to each user, several approaches were proposed. The proposed system efficiently store videos in the clouds and elastically constructs subVC (private agent) for each active mobile user, which try to offer "non-terminating" multimedia streaming services by adapting to the fluctuating link quality based on scalable video coding technique. And also try to offer "non-buffering" multimedia streaming experience by background prefetching based on the interactions of the users in SNSs. This paper mainly focused on verifying that how cloud computing can enhance the transmission adaptability and scalability as well as the prefetching the multimedia contents in advanced for mobile users. The proposed approach will enhance the QoS of multimedia streaming and gives better QoE to users.

In the future, we will try to improve the security issues in the proposed cloud architecture and SNS-based prefetching. And also try to carry out the large-scale implementation with the serious attention on energy and price cost.

ACKNOWLEDGMENT

I hereby take this opportunity to express my heartfelt gratitude towards the people whose help was very useful for the completion of my research work on the topic of " Social-aware video pre-fetching for multimedia streaming" It is my privilege to express sincerest regards to the project Guide Dr. N. J. Uke, for his valuable inputs, able guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of my project work. I deeply express my sincere thanks to our Co-guide Mr. N. R. Gavai for his encouragement and support.

REFERENCES

[1] CISCO, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011–2016," Technical Report, 2012.

[2] Y. Li, Y. Zhang, and R. Yuan, "Measurement and analysis of a large scale commercial mobile Internet TV system," in Proc. ACM Internet Meas. conf., 2011, pp. 209–224.

[3] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, and P. Amon, "Real-time system for adaptive video streaming based on SVC," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 9, pp. 1227–1237, Sep. 2007.

[4] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," IEEE Trans. Circuits and Systems for Video Tech., Vol. 17, No. 9, pp. 1103–1120, 2007.

[5] Z. Huang et al., "CloudStream: Delivering High-Quality Streaming Videos through A Cloud-based SVC Proxy," Proc. IEEE INFOCOM, 2011, pp. 201–205.

[6] T. Rodrigues et al., "On Word-of-Mouth Based Discovery of the Web," Proc. ACM IMC, 2011.

[7] A. Zambelli, "IIS smooth streaming technical overview," Microsoft Corp., 2009.

[8] Y. Fu, R. Hu, G. Tian, and Z. Wang, "TCP-friendly rate control for streaming service over 3G network," in Proc. WiCOM, 2006, pp. 1–4.

[9] K. Tappayuthpijarn, G. Liebl, T. Stockhammer, and E. Steinbach, "Adaptive video streaming over a mobile network with TCP-friendly rate control," in Proc. IWCMC, pp. 1325–1329, 2009.

[10] V. Singh and I. D. D. Curcio, "Rate adaptation for conversational 3G video," in Proc. IEEE INFOCOM Workshop, pp. 205–211, 2009.

[11] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in Proc. ACM MMSys, pp. 157–168, 2011.

[12] E. Piri, M. Uitto, J. Vehkaper, and T. Sutinen, "Dynamic cross-layer adaptation of scalable video in wireless networking," in Proc. IEEE GLOBECOM, pp. 1–5, 2010.

[13] P. McDonagh, C. Vallati, A. Pande, and P. Mohapatra, "Quality-oriented scalable video delivery using H. 264 SVC on an LTE network," in Proc. WPMC, 2011.

[14] Bharath, V., and K. Priyadarsini. "Energy Efficient Data Traffic Management in Mobile Cloud Computing" ENERGY 2, no. 4, 2014.

[15] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," J. Internet Services Applic., vol. 1,no. 1, pp. 7–18, Apr. 2010.

[16] D. Niu, H. Xu, B. Li, and S. Zhao, "Quality-assured cloud bandwidth auto-scaling for video-on-Demand applications," in Proc. IEEE INFOCOM, pp. 460–468, 2012.

[17] F. Wang, J. Liu, and M. Chen, "CALMS: Cloud-assisted live media streaming for globalized demands with time/region diversities," in Proc. IEEE INFOCOM, pp. 199–207, 2012.

[18] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: Architecture, applications, and approaches," Wiley J. Wireless Commun. Mobile Computing, Oct 2011.

[19] S. Chetan, G. Kumar, K. Dinesh, K. Mathew, and M. A. Abhimanyu, "Cloud Computing for Mobile World," 2010.

[20] N. Davies, "The case for VM-Based Cloudlets in mobile computing," IEEE Pervasive Computing, vol. 8, no. 4, pp. 14–23, 2009.

[21] B. Aggarwal, N. Spring, and A. Schulman, "Stratus: Energy-efficient-mobile communication using cloud support," in Proc. ACM SIGCOMM, pp. 477–478, 2010.

[22] Y. Zhang, W. Gao, G. Cao, T. L. Porta, B. K, and A. Iyengar, "Social-aware data diffusion in delay tolerant MANET," in Handbook of Optimization in Complex Networks: Commun. and Social Networks. Berlin, Germany: Springer, 2010.

[23] Z. Wang, L. Sun, C. Wu, and S. Yang, "Guiding Internet-scale video service deployment using microblog-based prediction," in Proc. IEEE INFOCOM Mini-conf., pp. 2901–2905, 2012.

[24] C. Ji and J. Zhou, "A study on recommendation features for an RSS reader," in Proc. Int. Conf. CyberC, pp. 193–198, Oct. 2010.

[25] J. J. Samper, P. A. Castillo, L. Araujo, J. J. Merelo, Cordón Ó., and F. Tricas, "NectaRSS, an intelligent RSS feed reader," J. Netw. Comput. Appl., vol. 31, no. 4, pp. 793–806, Nov. 2008.

[26] A. Das, M. Datar, and A. Garg, "Google news personalization: Scalable online collaborative filtering," in Proc. 16th Int. Conf. World Wide Web, pp. 271–280, 2007.

[27] G. Cugola and H.-A. Jacobsen, "Using publish/subscribe middleware for mobile systems,"ACM SIGMOBILE Mobile Comput. Commun. Rev., vol. 6, no. 4, pp. 25–33, Oct. 2002.

[28] Y. Kim, J. W. Lee, S. R. Park, and B. C. Choi, "Mobile advertisement system using data push scheduling based on user preference," in Wireless Telecommun. Symp., pp. 1–5, 2009.

[29] S. Zhao, P. P. C. Lee, J. C. S. Lui, X. Guan, X. Ma, and J. Tao, "Cloud based push-styled mobile botnets: A case study of exploiting the cloud to device messaging service," inProc. ACSAC, pp. 119–128, 2012.

[30] G. Wang and T. E. Ng, "The impact of virtualization on network performance of amazon EC2 data center," inProc. IEEE INFOCOM, 2010, pp. 1163–1171.

[31] A. Balasubramanian, R. Mahajan, and A. Venkataramani, "Augmenting mobile 3G using WiFi," inProc. ACM MobiSys, 2010, pp. 209–222.

[32] F. Benevenuto, T. Redrigues, V. Almeida, and J. Almeida, "Video interactions in online social networks," ACM Trans. Multimedia Cmputing, Commun. Applic., vol. 5, no. 4, pp. 30–44, 2009.

[33] J. M. Kang, S. S. Seo, and J. W. Hong, "Personalized battery lifetime prediction for mobile devices based on usage patterns,"J. Computing Sci. Eng., vol. 5, no. 4, pp. 338–345, 2011.

[34] M. Cha, H. Kwak, P. Rodriguez, Y. Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," inProc. ACM Internet Meas. Conf., 2007, pp. 1–14.